Search Engine Technology and Digital Libraries

Moving from Theory to Practice

Friedrich Summann
Bielefeld University Library, Germany
<summann@ub.uni-bielefeld.de>

Norbert Lossau
Bielefeld University Library, Germany
<lossau@ub.uni-bielefeld.de>

Abstract

This article describes the journey from the conception of and vision for a modern search-engine-based search environment to its technological realisation. In doing so, it takes up the thread of an earlier article on this subject, this time from a technical viewpoint. As well as presenting the conceptual considerations of the initial stages, this article will principally elucidate the technological aspects of this journey.

The conception of an academic search engine

The starting point for the deliberations about development of an academic search engine was the experience we gained through the generally successful project "Digital Library NRW", in which from 1998 to 2000—with Bielefeld University Library in overall charge—we designed a system model for an Internet-based library portal with an improved academic search environment at its core. At the heart of this system was a metasearch with an availability function, to which we added a user interface integrating all relevant source material for study and research. The deficiencies of this approach were felt soon after the system was launched in June 2001. There were problems with the stability and performance of the database retrieval system, with the integration of full-text documents and Internet pages, and with acceptance by users, because users are increasingly performing the searches themselves using search engines rather than going to the library for help in doing searches. Since a long list of problems are also encountered using commercial search engines for academic use (in particular the retrieval of academic information and long-term availability), the idea was born for a search engine configured specifically for academic use. We also hoped that with one single access point founded on improved search engine technology, we could access the heterogeneous academic resources of subject-based bibliographic databases, catalogues, electronic newspapers, document servers and academic web pages.

Software evaluation and technical realisation

Following on from our fundamental deliberations about an academic search engine, we searched the market for suitable software products. Our discussions with Google in 2002 broke down at an early stage, as we were only able to speak to sales personnel, and at that time at least, we received no indication that we could install Google software for testing locally. We found the situation different with the search engine Convera, and we were able to install their search engine on a machine in Bielefeld and test it for a limited period. We spent two weeks intensively observing the Convera software and concluded that it was more appropriate for an intranet installation than it was for our application of it as an Internet search engine.

We also tested the Russian open source search engine MnoGo and found many positive aspects to that software, but in our tests we also encountered performance problems when trying to process large amounts of data. Finally, we contacted the Norwegian software company Fast, which in 2002 was one of the market leaders alongside Google with the Fast search engine Alltheweb. A test installation was quickly and flexibly agreed, the technical realisation of which also succeeded smoothly and without any problems. Our experience with the Fast software was so positive that by the end of the test period, it was clear that we should carry out a proof-of-concept with this search

engine. Within the context of the proof-of-concept we aimed to bring together and make available a representative and heterogeneous amount of academic on-line material. Various document types and formats (both full-text and metadata), and the contents of the visible and invisible web were to be included. As a basic condition, we emphasised that the test should be carried out under live conditions on the basis of Fast Data Search. In addition, interoperability standards (OAI, XML) should be followed and prototypes of an intelligent and flexible user interface developed.

The technical work in Bielefeld University Library began in earnest in the summer of 2003. The technical core team has consisted of two software developers, who have been creating prototypes based on the FAST Data Search software. A concrete start was made with the realisation of a "Math Demonstrator", which—through its concentration on a single subject area—should form the basis for further development and discussion. In spring 2004 this subject based approach was extended with the aim to create a general "Digital Collections Demonstrator". Both demonstrators had their public launch in June 2004 with the establishment of the Bielefeld Academic Search Engine (BASE, http://base.ub.uni-bielefeld.de/). The work in Bielefeld has also formed the base for a collaborative project proposal "Search engine technology" to the Deutsche Forschungsgemeinschaft by Bielefeld University Library and the Regional Service Centre for Academic Libraries in North Rhine-Westphalia (HBZ, Cologne) as part of the Distributed Document Server (VDS) initiative [1]. This joint project is one way of extending the activities in Bielefeld and is expected to kick off at the end of 2004.

Technical details of the search engine solution

The technical structure of the FAST search engine is modular and transparent in construction and includes the standalone system components of a back-end and a front-end server. At the moment, Bielefeld is running one front-end server, but it could easily run more. Likewise the number of back-end servers is scaleable, and both areas can be rebuilt to create a multi-node system without creating any problems.

The front-end handles the tasks of providing the search environment, results analysis and presentation, and at the moment it is running on a Linux PC with 2 processors under SUSE 9.0. Connection to the web is established using PHP 4 on an Apache Web Server. The back-end deals with the areas of data loading, pre-processing and data conversion, data assimilation, crawling and document processing and indexing.

The user interface is bi-lingual (German and English). Next to the basic search form with Google-like single line search boxes is an advanced search option (see Figure 1) with additional functionality, which is the focus of the software development. It is here that the supplementary functions, such as refined and restricted searching, choice of collections and search history, are offered. Both search forms allow the search to be restricted to documents that are freely available.

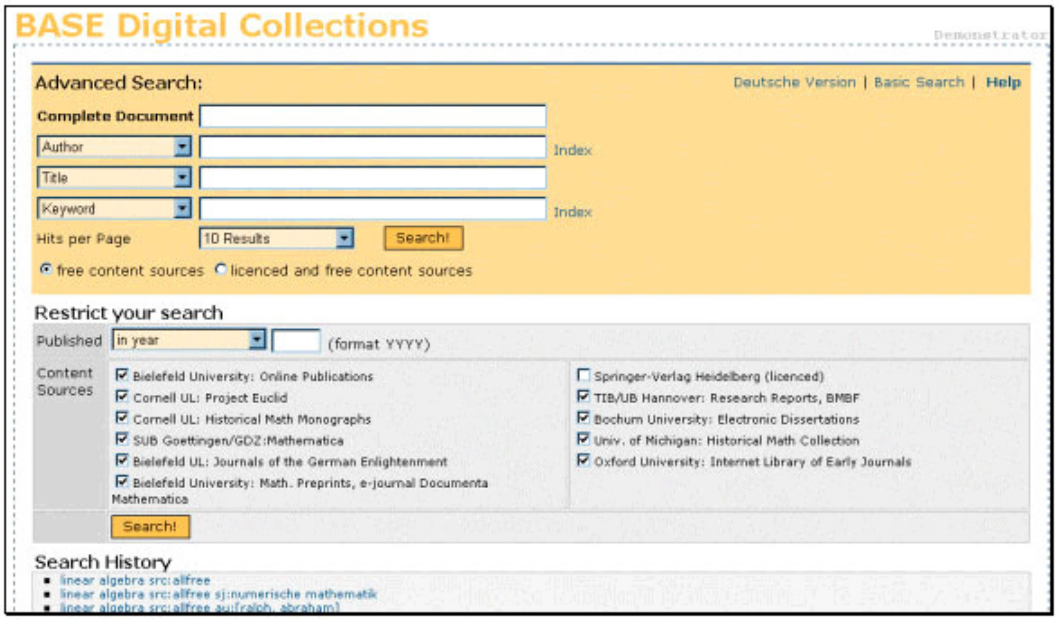

Figure 1: Advanced search screen

The results page (Figure 2) differs from the search engine standard in that it shows a sophisticated display of metadata whenever metadata is present in the document. Next to the displayed search results there is an interactive area where the user can refine the search (for example, by using the metadata to search for author and classification and for formal aspects such as document format and collection). The appropriate fields from all the results are formed into a dropdown menu. It is possible to do further searches with respect to a single document by searching for similar documents in the master index (find similar), within the search results (refine similar) or by excluding similar documents in the search results (exclude similar). The search history option completes the current capabilities of the user interface.

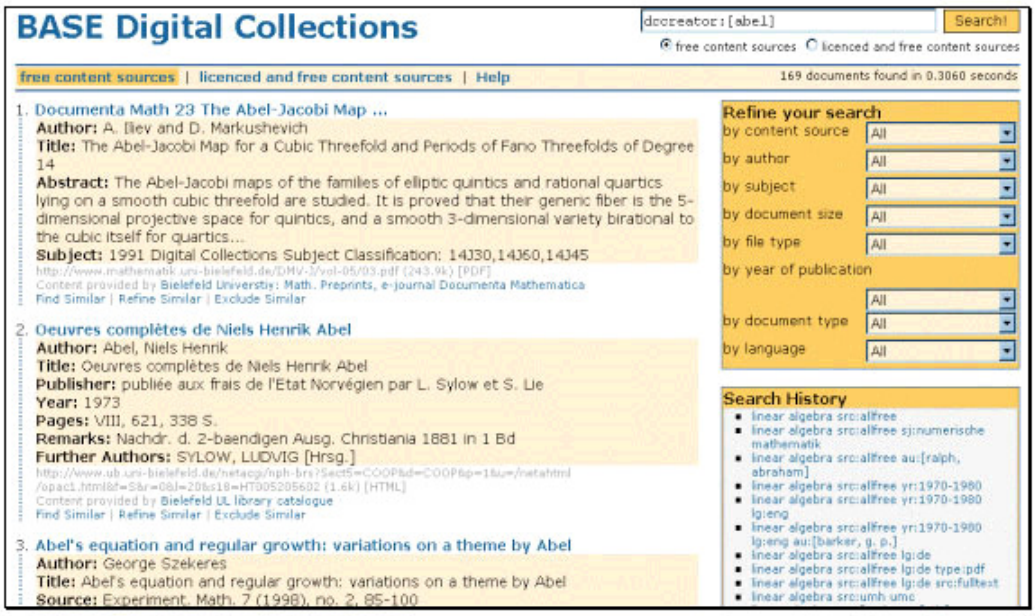

Figure 2: Results page

The back-end system is a live system running on a Linux PC under SUSE Linux 9.0 with two processors and a RAID system with a hard disk capacity of 290GB, and in parallel a test system is running, likewise on a Linux PC, that allows internal alterations to be made for development purposes without affecting the live system.

The contents of the collection

As of June 2004, approximately 600,000 documents had been captured and distributed in 15 collections. The back-end server requires about 25GB free space for this. The choice of captured sources was made with the aim of capturing representative data of different types. In doing so, the following were processed:

a) Metadata

Euclid Project (OAI) (6516 articles)
Bielefeld Library Catalogue (copy of database) (70,000 records)
Zentralblatt Math (copy of database)
Articles from the project "Enlightenment Newspapers" (Zeitschriften der Aufklärung) (copy of database) (57690 Articles)

b) Full-text without metadata

Documenta Mathematica (open access e-journal)
Bielefeld University Pre-print server of Mathematics (Crawling) (18,000 documents)
Project reports for the Federal State Ministry for Education and Research (Crawling) (64 documents)

c) Full-text with metadata

Springer newspapers (224,382 articles)
Bochum University Library Hochschulschriftenserver (OAI-Harvesting) (1,908 documents)
Bielefeld University Library Hochschulschriftenserver (OAI-Harvesting) (369 documents)
Internet Library of Early Journals at Oxford (SGML Export) (104,516 pages)
University of Michigan Historical Math Collection (OAI Harvesting) (772 documents)
Cornell University Library Historical Math Monographs (OAI Harvesting) (360 documents)
Göttingen State and University Library Mathematica (OAI Harvesting) (427 documents)

The integration of data sources

In the sphere of the metasearch, which currently forms the basic search of the library portal in Bielefeld, work towards the integration of further resources consists of transmitting connection information to the target system (Z39.50 or http-based), implementing database queries based on this and transforming the delivered results into an internal format.

With the use of search engines, a completely different approach is necessary. It is possible to load data into the index in a number of ways. FAST offers three service interfaces for data capture: Webcrawler, Database Connector (for access to relational databases such as Oracle etc.) and File Traverser (access to data). So far, Database Connector has not been needed within the framework of BASE, since no data of this sort has had to be included. Figure 3 gives an overview of the dataflow with the main stages.

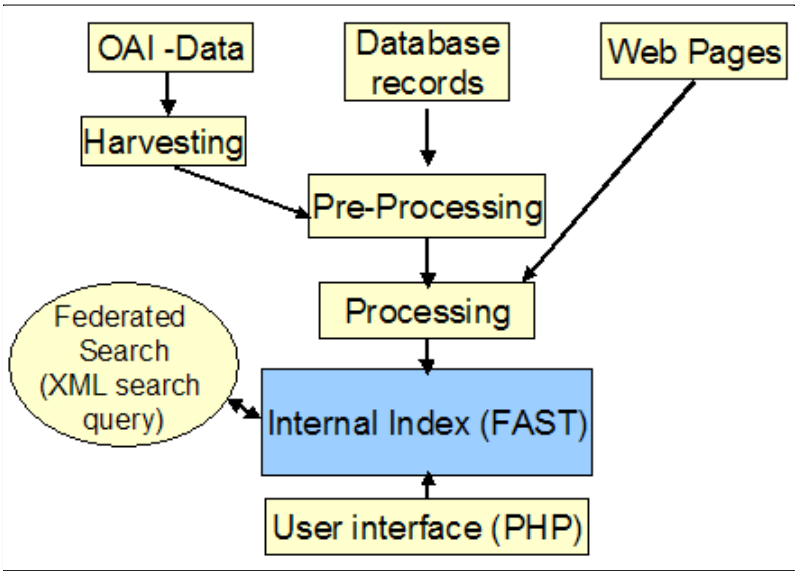

Figure 3: Dataflow BASE

In order to process documents that are not encoded in HTML, we used the FAST tool File Traverser. This transformed the existing proprietary format into a FAST-defined XML format during pre-processing. An essential emphasis of the development was the harvesting of OAI data, through which, with the help of an open-source product from Virginia Polytechnic Institute and State University, these data were collected and stored. In practice, however, a catalogue of problems arose during the analysis of OAI data and these problems will require flexible and configurable solutions. The Dublin Core format used by the OAI introduced very heterogeneous formats into the data field or the language code, which then had to be individually investigated. The URL leading to the full text was sometimes placed in the source-field, and these constellations had to be taken into account. Therefore, we are striving for extensive software solutions in this sphere by further developing the Perl and XSLT scripts written so far to include universal configuration functions. All in all, these applications in the pre-processing stage will complete the following tasks: language code recognition, date-sorting, XML conversion, creation of a unique identifier, general filtering and error correction, creation of element values, and establishment of full-text links.

In the area of processing, however, internal and local filter programs that can alter the data being processed will be defined for the internal data import process. Tasks undertaken by File Traverser, such as Language recognition, Mime-type recognition, Teaser generation and Field ordering, belong

with these programs. With crawling, among others, the following tasks are carried out: format recognition, decompression, setting of the internal content type (full-text, metadata, or a mixture of both), format conversion from Postscript or PDF, and language determination within the framework of processing.

One particular task consists of bringing together metadata and full-text information in a single data record, which then can be processed as a unit for searching and presentation of results.

The FAST module used for the data capture process as well as the additional module and tools developed are listed in Table 1. As also can be seen in Table 1, the in-house developments have all originated with open source solutions.

	FAST	**Additional developments**
Data Loading	Crawler, File Traverser, DB Connector	OAI Harvester DB export
Pre-processing		Perl, XSLT crosswalks
Processing	Standard stages	Python stages
Indexing	Indexer	
Access, Navigation	Search API	PHP scripts

Table 1: Overview of tools used

In the FAST system, an index structure defined for BASE has been designed that contains the fundamental 15 Dublin Core fields. In addition, currently 5 extra fields have been defined that are intended to contain ISBN/ISSN, DOI, year (normalised form), source type (metadata, full-text, etc.) and source.

Further developments and vision

Much development of the front-end remains to be done. The flexible integration of the search engine in external surroundings should be supported by the introduction of template technology in order to make local views of the search engine possible through establishment of search and result parameters. It is already possible to add one or more search boxes to any web page, with which the BASE search engine can be integrated in external portal surroundings. A further development in this area would then support a differentiation in the results page. In addition, the search interface should be developed on the basis of the Search API, and the results page should show a combination of metadata results and the appropriate full-text.

For the back-end, the main development emphasis will be the automation and configurability of the harvesting and pre-processing of the documents, in particular to come to grips with the problems we experienced with OAI harvesting that we mentioned earlier in this article. The improvement of search results (ranking, boosting, linguistic methods) must undergo fine tuning from the point of view of academic use. Improving performance is another important goal. Since we want the search engine to be able to provide basic services using external systems and portals, we plan to implement standard interfaces (Z39.50, OAI, SOAP). For collaboration with other systems, we plan to activate features enabling distributed search and connection with external indexes.

---

[1] This initiative is carried by the Working Committee of German Library Service Centres.

doi:10.1045/september2004-lossau